CLAIM AMENDMENTS

This listing of claims will replace all prior versions, and listings, of claims in the application.

1. (Currently Amended) A method for reconfiguring thread scheduling ~~to run in parallel with a main processor,~~ comprising the steps of:

obtaining parameter values by monitoring memory operations for a plurality of different threads, the parameter values each modified by execution of a corresponding one or more of the threads by a processor;

determining if thread scheduling is to be reconfigured by performing logic functions on the parameter values using a thread scheduler function unit that performs said determining[[,]] in parallel with, but without interrupting execution by the ~~main~~ processor of a currently enabled one of the threads~~, on said parameter values to determine if thread scheduling should be reconfigured~~, and, if so, which one of the threads should be enabled; and

sending an interrupt signal to interrupt the ~~main~~ processor after determining that ~~if~~ thread scheduling is to be reconfigured.

2. (Original) The method as defined in claim 1, wherein said performing logic functions step is performed on a continuous basis.

3. (Original) The method as defined in claim 1, wherein the obtaining parameter values step comprises monitoring the values from thread processes held in memory mapped registers with fixed addresses.

4. (Original) The method as defined in claim 1, wherein said performing logic functions step comprises performing said logic functions substantially simultaneously on a substantial plurality of said parameter values.

5. (Original) The method as defined in claim 1, where the performing logic functions step is not performed by a microprocessor, but rather by hardware logic.

6. (Original) The method as defined in claim 1, wherein the performing logic functions step comprises performing logic functions on reconfigurable hardware.

7. (Original) The method as defined in claim 1, further comprising the step of, during the performing logic functions step, receiving at least one additional parameter value; and performing logic functions with said at least one additional parameter value to determine if thread scheduling should be reconfigured.

8. (Original) The method as defined in claim 1, wherein the performing logic functions step comprises performing said logic functions with a microengine or microprocessor.

9. (Currenty Amended) The method as defined in claim 8, further comprising the step of, during performing logic functions step in the microengine or microprocessor, receiving at least one additional parameter value; and when the microengine or microprocessor is free, performing logic functions with said at least one additional parameter value to determine if thread rescheduling should be reconfigured.

10. (Original) The method as defined in claim 9, wherein the performing logic functions with the at least one additional parameter comprises performing a different logic function as compared to an immediately preceding logic function performed in the performing logic functions step.

11. (Original) The method as defined in claim 1, wherein local copies of the parameter values are held in a set of registers, and wherein said obtaining step comprises snooping memory operations for data addressed to a plurality of predetermined locations and updating the local copies thereof in the set of registers.

12. (Currently Amended) The method as defined in claim 11, wherein said snooping memory operations include memory operations for the ~~main~~ processor and memory operations of other processors in a multiprocessor system.

13. (Original) The method as defined in claim 11, further comprising the step of, receiving at least one additional parameter value during the performance of the performing logic functions step; and when the initial performance of the performing logic functions step is completed, performing logic functions with said at least one additional parameter value to determine if thread rescheduling should be reconfigured.

14. (Currently Amended) ~~A method~~ The method as defined in claim 1, wherein one of said parameter values is a time devoted to a currently running thread.

15. (Currently Amended) ~~A method~~ The method as defined in claim 1, wherein one of said parameter values is an amount of data that a predetermined queue is able to produce.

16. (Currently Amended) ~~A method~~ The method as defined in claim 1, wherein one of the parameter values is an amount of data that may be consumed by a predetermined queue.

17. (Currently Amended) The method as defined in claim 1, wherein the thread scheduling function and the function of the ~~main~~ processor are performed on a single chip.

18. (Original) The method as defined in claim 1, wherein the performing logic step comprises storing interim and or final results from the performing logic step.

19. (Original) The method as defined in claim 1, wherein the performing logic functions step includes the step of determining when a

parameter value for a thread has been modified and determining an identity of the parameter that has been modified; and, wherein said performing logic functions step is performed with said identity of the modified parameter used, in part, to pick a specific logic function to perform.

20. (Currently Amended) A system for processing, including a parallel hardware thread scheduler, comprising:

a ~~main~~ processor for executing threads;

a plurality of memory mapped registers for monitoring memory operations for a plurality of the threads, each of said registers holding a different thread parameter which is modified by execution of a corresponding one or more of the threads;

reconfigurable hardware logic connected to receive a ~~substantial~~ plurality of outputs from said registers in parallel and to perform logic functions substantially simultaneously thereon, in parallel with, but without interrupting the ~~main~~ processor, to determine if thread scheduling should be reconfigured, and if so, determining which thread should be enabled; and

a circuit for sending an interrupt signal to interrupt the ~~main~~ processor if thread scheduling is to be reconfigured.

21. (Currently Amended) A system for processing, including a parallel hardware thread scheduler, comprising:

a ~~main~~ processor for executing threads;

a hardware snooping logic detecting from memory traffic selected addresses for parameter values for a plurality of ~~different~~ the threads, the parameter values each modified by execution of a corresponding one or more of the threads, including a set of registers for holding local copies of said parameter values with said selected addresses, and logic for updating one of said local copies when the address therefor has been detected;

reconfigurable hardware logic connected to receive a ~~substantial~~ plurality of outputs from said registers in parallel and to perform logic functions substantially simultaneously thereon, in parallel with, but without interrupting the ~~main~~ processor, to determine if thread scheduling should be reconfigured, and if so, determining which thread should be enabled; and

a circuit for sending an interrupt signal to interrupt the ~~main~~ processor if thread scheduling is to be reconfigured.

22. (Currently Amended) A system for reconfiguring thread scheduling ~~to run in parallel with a main processor,~~ comprising:

a first component for obtaining parameter values by monitoring memory operations for a plurality of different threads, the parameter values each modified by execution of a corresponding one or more of the threads by a processor;

a second component comprising a thread scheduler function unit for performing logic functions, in parallel with, but without interrupting execution by a ~~the main~~ processor of a currently enabled one of the threads, on said parameter values to determine if the thread scheduling on the ~~main~~ processor should be reconfigured, and if so, which thread should be enabled; and

a third component for sending an interrupt signal to interrupt the ~~main~~ processor ~~if~~ after determining that thread scheduling is to be reconfigured.

23. (Currently Amended) A system for reconfiguring thread scheduling ~~to run in parallel with a main processor,~~ comprising:

reconfigurable hardware for obtaining parameter values for a plurality of different threads, the parameter values each modified by execution of a corresponding one or more of the threads by a processor;

logic for performing first logic functions, in parallel with, but without interrupting the ~~main~~ processor, on said parameter values to determine if the thread scheduling on the ~~main~~ processor should be reconfigured and which thread should be enabled; and

logic for triggering ~~a second process to run~~ second logic functions to be performed after the first logic functions ~~process to perform second logic functions~~ to determine which thread should be enabled when at least one ~~second~~ parameter is updated during a period when the first logic functions are being performed ~~process is running~~.

24. (New) The method according to claim 1, further comprising obtaining at least one additional parameter value modified by execution of at least one additional processor and wherein the logic functions are performed on the at least one additional parameter value.

25. (New) The system according to claim 20, further comprising at least one additional processor that modifies at least one additional thread parameter value and wherein the logic functions are performed on the at least one additional thread parameter value.

26. (New) The system according to claim 22, further comprising at least one additional processor that modifies at least one additional parameter value and wherein the logic functions are performed on the at least one additional parameter value.

27. (New) The system according to claim 22, wherein the performing logic functions is performed on reconfigurable hardware.

28. (New) The system according to claim 22, wherein the performing logic functions is performed on a microengine or microprocessor.

29. (New) The system according to claim 23, further comprising at least one additional processor that modifies at least one additional parameter value and wherein the logic functions are performed on the at least one additional parameter value.